Dynamic Calibrated On-Chip Termination (ALTOCT) Megafunction

User Guide

101 Innovation Drive
San Jose, CA 95134
www.altera.com

UG-01003-3.0

Subscribe

Contents

1. About this Megafunction

The Dynamic Calibrated On-Chip Termination (ALTOCT) megafunction is used in double-data rate (DDR) external memory interfaces. This megafunction is closely associated with the External DDR Memory PHY Interface (ALTMEMPHY) megafunction. On-chip termination (OCT) improves signal quality over external termination through reduced parasitic, board space, and external component costs.

Features

The ALTOCT megafunction provides the following features:

- Support for up to 10 OCT blocks
- Support for calibrated on-chip series termination (RS) and calibrated on-chip parallel termination (RT) on all I/O pins
- Calibrated termination values of 25 and 50 ohm

Device Support

TheALTOCT megafunction supports the following Altera® devices:

- Arria® II GX
- Arria II GZ
- Arria V
- Cyclone® V
- HardCopy® III
- HardCopy IV
- Stratix® III
- Stratix IV
- Stratix V

2. Parameter Settings

This section describes the parameter settings for the ALTOCT megafunction.You can parameterize the megafunction using the MegaWizard™ Plug-In Manager or the command-line interface (CLI). Altera recommends that you configure the megafunctions using the MegaWizard Plug-In Manager.

This user guide assumes that you are familiar with megafunctions and how to create them. If you are unfamiliar with Altera® megafunctions, refer to the *Introduction to Megafunctions User Guide*.

MegaWizard Parameter Settings

Table 2–1 provides descriptions of the options available on the individual pages of the ALTOCT MegaWizard Plug-In Manager.

Table 2–1. ALTOCT MegaWizard Plug-In Manager Page Options and Description (Part 1 of 2)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
1	Which action do you want to perform?	You can select from the following options: **Create a new custom megafunction variation**, **Edit an existing custom megafunction variation**, or **Copy an existing custom megafunction variation**.
2a	Select a megafunction from the list below	Select **ALTOCT** from the **I/O** category.
	Which device family will you be using?	Specify the device family that you want to use.
	Which type of output file do you want to create?	You can choose AHDL(**.tdf**), VHDL(**.vhd**), or Verilog HDL (**.v**) as the output file type.
	What name do you want for the output file?	Specify the name of the output file.
	Return to this page for another create operation	Turn on this option if you want to return to this page to create multiple megafunctions.

Table 2–1. ALTOCT MegaWizard Plug-In Manager Page Options and Description (Part 2 of 2)

MegaWizard Plug-in Manager Page	Configuration Setting	Description
3	Currently selected device family	Specifies the device family you chose on page 2a.
	Match project/default	Turn on this option to ensure that the device selected matches the device family that is chosen in the previous page.
	Calibrate OCT on power-up	Turn on this option if you want to calibrate OCT on power-up.
	How many OCT blocks should be used?	Specify the number of OCT blocks for your design.
	Enable parallel termination	Turn on this option if you want to enable parallel termination instead of series termination. Observe the changes in resource usage when this option is enabled.
	Create 'calibration_wait' input port to prevent calibration	The `calibration_wait` input port can be used to halt calibration operation. This option is for advanced users only. Typical users should not enable this option.
	Create 'clken' input port	The `clken` input port is used as the clock enable signal. This option is for advanced users only. Typical users should not enable this option.
	Enable independent calibration/shift	Turn on this option to enable independent calibration/shift.
4	Generate netlist	Turn on this option if you want to generate a netlist for your third-party EDA synthesis tool to estimate the timing and resource usage of the megafunction. If you turn on this option, a netlist file (**_syn.v**) is generated. This file is a representation of the customized logic used in the Quartus® II software and provides the connectivity of the architectural elements in the megafunction but may not represent true functionality.
5	Summary Page	Specify the types of files to be generated. The Variation file (<*function name*>**.v**) contains wrapper code in the language you specified on page 2a and is automatically generated. Choose from the following types of files: ■ AHDL Include file (<*function name*>**.inc**) ■ VHDL component declaration file (<*function name*>.**cmp**) ■ Quartus II symbol file (<*function name*>**.bsf)** ■ Instantiation template file (<*function name*>**_inst.v**) ■ Verilog HDL black box file (<*function name*>**_bb.v)** For more information about the wizard-generated files, refer to the Quartus II Help or to the *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*.

Command Line Interface Parameters

Expert users can choose to instantiate and parameterize the megafunction through the command-line interface using the clear box generator command. This method requires you to have command-line scripting knowledge.

For more information about using the clear box generator, refer to the *Introduction to Megafunctions User Guide*.

Table 2–2 lists the parameters for the ALTOCT megafunction.

Table 2–2. ALTOCT Megafunction Parameters

Parameter Name	Type	Required	Description
ENABLE_PARALLEL_TERMINATION	String	Yes	Enables the RT State Machine. Values are **TRUE** and **FALSE**. If omitted, value is **FALSE**.
OCT_BLOCK_NUMBER	Integer	Yes	The number of OCT blocks in the design. Values are integer 1 to 10.
LPM_HINT	String	No	Allows you to specify Altera-specific parameters in VHDL Design Files (**.vhd**). The default value is **UNUSED**.
LPM_TYPE	String	No	Identifies the library of parameterized modules (LPM) entity name in VHDL Design Files (**.vhd**).
INTENDED_DEVICE_FAMILY	String	No	This parameter is used for modeling and behavioral simulation purposes. Create the ALTOCT megafunction with the MegaWizard Plug-In Manager to get the value for this parameter.

3. Functional Description

This chapter describes the functional description and the design examples of the ALTOCT megafunction. This section also includes the ports descriptions of the ALTOCT megafunction. You can use the ports to customize the ALTOCT megafunction according to your application.

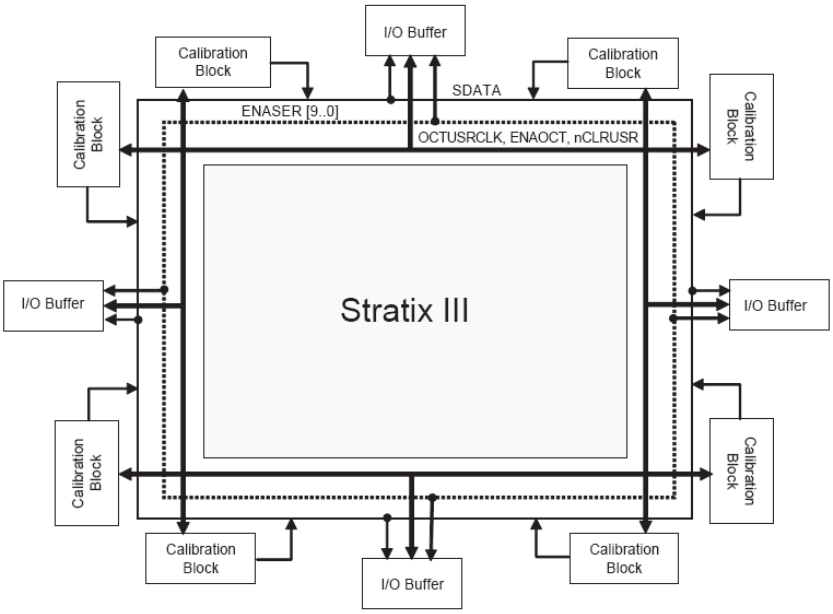
OCT Architecture

The OCT calibration architecture provides dynamic series and parallel on-chip termination to improve I/O impedance matching and termination capabilities. OCT improves signal quality over external termination through reduction of parasitic, board space, and external component costs. The OCT architecture support RS with and without calibration, RT with calibration, dynamic series and parallel termination for single-ended I/O standards, and on-chip differential termination (RD) for differential LVDS I/O standards. OCT is supported in all I/O banks by selecting one of the OCT I/O standards.

Figure 3–1 shows the OCT calibration architecture in Stratix III devices. The SDATA, OCTUSRCLK, and ENASER signals are used to serially transfer calibrated codes from each OCT calibration block to any I/O. To serially shift the 14-bit OCT RS calibration code and the 14-bit OCT RT calibration code into the registers located in the I/O buffer, 28 clock cycles using OCTUSRCLK are required. When calibration is complete, the 28-bit OCT calibration code (14-bit OCT RS code and 14-bit OCT RT) must be serially shifted out from each OCT calibration block to the corresponding I/O buffer.

After calibrated codes are shifted in serially to each I/O bank, the calibrated codes must be converted from serial format to parallel format before the codes are used in the I/O buffers. Use the S2PENA signals to complete the serial-to-parallel shifting.

Figure 3–1. Signals for Shift-Out Codes from the OCT Calibration Block to I/O Buffers



Power-Up Mode Calibration and On-Demand Calibration in Quartus II Software

Stratix III and Stratix IV have two termination related assignments: INPUT_TERMINATION and OUTPUT_TERMINATION. Termination can exist on input and output buffers, and sometimes simultaneously.

When calibrated termination uses only the Quartus® Settings File (QSF) assignments, the power-up mode of the calibration scheme is used, and on-demand calibration updates are unavailable.

To use on-demand calibration, the ALTOCT megafunction must be instantiated into the design. If more than one group of pins needs to be calibrated on-demand, more than one calibration block must be instantiated.

There are two methods to associate pin groups with a calibration block:

- Instantiate the I/O buffer primitives at the top level and connect them to the appropriate calibration blocks.

All I/O banks with the same V_{CCIO} can share a calibration block, even if that particular I/O bank has its own calibration block. You can connect any number of I/O pins that support calibrated termination to a calibration block, unless the specific I/O pins on the device are not usable with a particular calibration block. The Quartus II software produces warning messages if there is no pin connected to the block.

- Use a QSF assignment to indicate which pin (bus) is associated with which calibration block.

For more information about the OCT architecture and the calibration modes for other device families, refer to the respective device handbooks.

Design Example: Series Calibration for Four Calibration Blocks Using Stratix III Device

This design example uses the ALTOCT megafunction to calibrate four calibration blocks using series termination for the Stratix III device. This example uses the MegaWizard Plug-In Manager in the Quartus II software.

In this example, you perform the following tasks:

- Generate the dynamic on-chip termination calibration blocks using the ALTOCT megafunction in the MegaWizard Plug-In Manager
- Ensure that correct termination assignments are used in the design by using the Assignment Editor
- Simulate the design in the ModelSim-Altera software

The design examples are available for download from the following locations:

- On the Documentation: Quartus II Development Software page, expand the **Using Megafunctions** section and then expand the **I/O** section.
- On the Documentation: User Guides section of the Altera website.

Generate the Dynamic On-Chip Termination Calibration Blocks

Perform the following steps to generate the dynamic on-chip termination calibration blocks:

1. Open **altoct_DesignExample.zip** and extract **altoct_ex.qar**.
2. In the Quartus II software, open **altoct_ex.qar** and restore the archive file into your working directory.
3. Open the top-level file, **altoct_ex.bdf**.
4. Double-click on a blank area in the schematic.
5. In the Symbol window, click **MegaWizard Plug-In Manager**. Page 1 of the MegaWizard Plug-In Manager appears.
6. Select **Create a new custom megafunction variation**.
7. Click **Next**. Page 2a of the MegaWizard Plug-In Manager appears.

8. In the MegaWizard Plug-In Manager pages, select or verify the configuration settings shown in Table 3–1. Click **Next** to advance from one page to the next.

Table 3–1. Configuration Settings for ALTOCT Design Example

MegaWizard Plug-in Manager Page	Configuration Setting	Value
2a	Select a megafunction	ALTOCT
	Which device family will you be using?	Stratix III
	Which type of output file do you want to create?	Verilog HDL
	What name do you want for the output file?	cal_out
3	Currently selected device family	Stratix III
	Match project/default	Turned on
	Calibrate OCT on power-up	Turned off
	How many OCT blocks should be used?	4
	Enable parallel termination	Turned off
	Create 'calibration_wait' input port to prevent calibration	Turned off
	Create 'clken' input port	Turned off
	Enable independent calibration/shift	Turned off
4	Generate netlist	Turned off
5	Variation file	Turned on
	Quartus II symbol file	Turned on
	Instantiation template file	Turned on
	Verilog HDL black-box file	Turned on
	AHDL Include file	Turned on
	VHDL component declaration file	Turned on

9. Click **Finish**. The `cal_out` module is now built.
10. In the Symbol window, click **OK**.
11. Move the mouse to align the cal_out symbol with the existing ports in the **altoct_ex.bdf** file. Click to place the symbol. You have now completed the design file.
12. On the File menu, click **Save**.

Verify Termination or Calibration Assignments of the Design Example

This section describes the necessary assignments for the design example. Perform the following steps to verify the assignments.

1. In the Quartus II software, on the Assignments menu, click **Assignment Editor**. The Assignment Editor appears.

   Figure 3–2 shows the assignments used in the design example.

Figure 3–2. Assignments Used in the altoct_ex Design

	From	To	Assignment Name	Value	Enabled
1		test_output[0]	I/O Standard	SSTL-18 Class I	Yes
2		test_output[1]	I/O Standard	SSTL-18 Class I	Yes
3		test_output[2]	I/O Standard	SSTL-18 Class I	Yes
4		test_output[3]	I/O Standard	SSTL-18 Class I	Yes
5		test_output[0]	Output Termination	Series 50 Ohm with Calibration	Yes
6		test_output[1]	Output Termination	Series 50 Ohm with Calibration	Yes
7		test_output[2]	Output Termination	Series 50 Ohm with Calibration	Yes
8		test_output[3]	Output Termination	Series 50 Ohm with Calibration	Yes
9		test_output[0]	Termination Control Block	cal_out:inst|cal_out_alt_oct_q8n:cal_out_alt_oct_q8n_component|sd1a_0	Yes
10		test_output[1]	Termination Control Block	cal_out:inst|cal_out_alt_oct_q8n:cal_out_alt_oct_q8n_component|sd1a_1	Yes
11		test_output[2]	Termination Control Block	cal_out:inst|cal_out_alt_oct_q8n:cal_out_alt_oct_q8n_component|sd1a_2	Yes
12		test_output[3]	Termination Control Block	cal_out:inst|cal_out_alt_oct_q8n:cal_out_alt_oct_q8n_component|sd1a_3	Yes

The first set of assignments are I/O Standard assignments that are used on the pins of the design. This selection depends on which I/O standards are valid for a particular application. In this design example, the functional output pins are `test_output[3..0]`, which are all assigned the SSTL-18 Class 1 standard.

The second set of assignments are Output Termination assignments that are assigned to pins that have termination. You can specify the assignments as non-calibrated termination or calibrated termination. In this design example, the functional output pins are `test_output[3..0]`, which are all assigned the Output Termination assignments with the value of **Series 50 Ohm with Calibration**. In this assignment, these pins function as outputs for the FPGA chip only.

The third set of assignments are Termination Control Block assignments that are used to indicate which calibration block is used to calibrate a particular I/O pin or group. In this design example, the functional output pins are `test_output[3..0]`, which are assigned with the value of `cal_out:inst|cal_out_alt_oct_q8n:cal_out_alt_oct_q8n_component|sd1a_x`, where `x` indicates the termination calibration block that is used. Becausethe output pins, `test_output[3..0]`, are four bits, each bit of the pin is assigned to one calibration block.

2. Verify that all the assignments in the Assignment Editor match what is shown in Figure 3–2.

Functional Simulation in the ModelSim-Altera Software

The design is simulated in the ModelSim®-Altera software to generate a waveform display of the device behavior. You should be familiar with the ModelSim-Altera software before using the design examples. To get started with the ModelSim-Altera software, refer to the ModelSim-Altera Software Support page on the Altera website. The support page includes links to such topics as installation, usage, and troubleshooting.

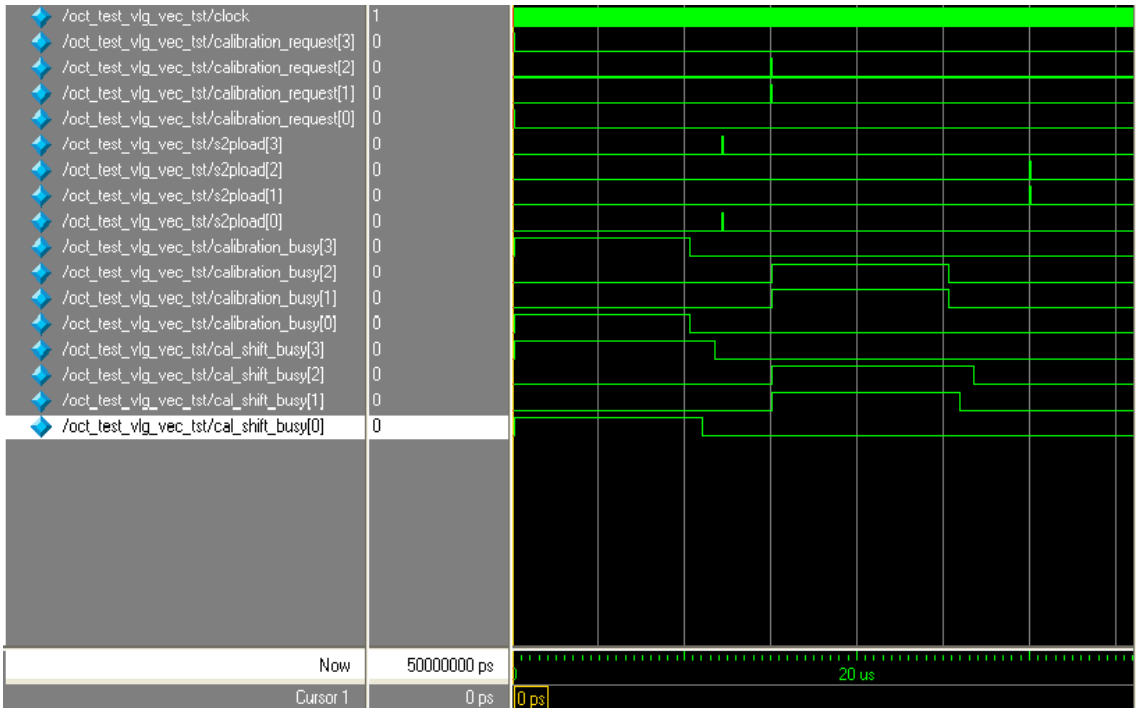
Set up and simulate the design in the ModelSim-Altera software by performing the following steps:

1. Unzip the **altoct_ex_msim.zip** file to any working directory on your PC.
2. Start the ModelSim-Altera software.
3. On the File menu, click **Change Directory**.
4. Select the folder in which you unzipped the files.
5. Click **OK**.
6. On the Tools menu, select **TCL**, then click **Execute Macro**.
7. Select the **altoct_ex_msim.do** file and click **Open**. The **altoct_ex_msim.do** file is a script file used by the ModelSim-Altera software to automate all the necessary settings for the simulation.
8. Verify the results shown in the Waveform Viewer window.

You can rearrange signals, remove signals, add signals, and change the radix by modifying the script in **altoct_ex_msim.do** accordingly.

Figure 3–3 shows the expected simulation results in the ModelSim-Altera software.

Figure 3–3. ModelSim-Altera Simulation Waveforms



For these functional results, the primary objective is to highlight the dynamic calibration process. The design example uses four calibration blocks named calibration block [x], with x=0, 1, 2, 3.

Each calibration block has its own set of control and data signals. For example, `calibration block[3]` has `calibration_request[3]`, `calibration_busy[3]`, and `cal_shift_busy[3]` signals.

To request calibration on `calibration block[x]`, the `calibration_request[x]` signal must be asserted for at least 1 clock cycle to initiate the calibration. Multiple calibration block requests are allowed.

The `calibration_busy[x]` signal indicates the calibration process that is being performed in `calibration block[x]`. Calibration typically takes 200 clock cycles. Calibration is complete when the `calibration_busy[x]` signal is deasserted.

The `cal_shift_busy[x]` signal is asserted at the same time as the `calibration_busy[x]` signal. This signal actually indicates both the duration of the calibration and the serial shifting of termination codes from `calibration block[x]` to the particular I/O buffers. Serial shifting of series termination codes typically takes 14 clock cycles.

After calibrated codes are shifted in serially to the particular I/O bank, the calibrated codes must be converted from serial format to parallel format before being used in the I/O buffers. To perform the conversion, the `s2pload[x]` signal is asserted at any time for 1 clock cycle after the shifting process.

In the simulation waveform shown in Figure 3–3, all of the calibration blocks have their calibration_request signals asserted at a particular time. Initially, requests for calibration are done simultaneously for calibration block[0] and calibration_request[3]. This can be observed in the behavior of the calibration_request[3] and calibration_request[0] signals. Both signals are asserted and deasserted at the same time.

Observe the behavior of the calibration_busy[3] and calibration_busy[0] signals. Notice that both are also asserted and deasserted at the same time. This is because multiple OCT calibration blocks can be calibrated at the same time. When these signals get deasserted, calibration is complete.

Next, observe the cal_shift_busy[3] and cal_shift_busy[0] signals. They are both asserted at the same time as the calibration_busy[3] and calibration_busy[0] signals, and both signals get deasserted after a number of clock cycles. These signals indicate the status of calibration and serial shifting of the termination codes.

Notice that the cal_shift_busy[3] and cal_shift_busy[0] signals get deasserted at different time—cal_shift_busy[0] gets deasserted first and followed by cal_shift_busy[3]. This is because the shifting process is serial, and only one calibration block can be active at a time. In the event of multiple calibration block requests, the priority calibration block[x] is based on this order. x=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, which means calibration block[0] shift codes first and then followed by the other blocks in the order. Observe Figure 3–3, the cal_shift_busy[0] gets deasserted first and then followed by cal_shift_busy[3]. This is because calibration block[0] has higher priority than calibration block[3].

Finally, the calibrated codes must be converted from serial format to parallel format before being used in the I/O buffers. This is done by asserting and deasserting the s2pload[0] and s2pload[3] signals for 1 clock cycle only after cal_shift_busy[3] and cal_shift_busy[0] signals have been deasserted. See Figure 3–3.

Similar calibration process as described on 3–7 applies to the remaining portion of the waveform. However, it applies to calibration block[2] and calibration block[1].

ALTOCT Megafunction Ports

Table 3–2 and Table 3–3 lists the input and output ports for the ALTOCT megafunction.

Input Ports

Table 3–2. ALTOCT Megafunction Input Ports

Port Name	Required	Description	Comments
aclr	No	Asynchronous clear	If omitted, value is **GND**
calibration_request	Yes	User request for calibration	Input port [OCT_BLOCK_NUMBER - 1..0] wide
calibration_wait	No	Clock cycles to wait before starting calibration after calibration request	Input port [OCT_BLOCK_NUMBER - 1..0] wide. If omitted, value is GND
clock	Yes	System clock	——
rdn	Yes	Pull-down reference resistor	Input port [OCT_BLOCK_NUMBER - 1..0] wide

Table 3–2. ALTOCT Megafunction Input Ports

Port Name	Required	Description	Comments
rup	Yes	Pull-up reference resistor	Input port [OCT_BLOCK_NUMBER - 1..0] wide
s2pload	Yes	Signal used to enable serial to parallel shifting of calibrated codes to I/O buffers	Input port [OCT_BLOCK_NUMBER - 1..0] wide
clken	No	Clock enable	If omitted, value is V_{cc}

Output Ports

Table 3–3. ALTOCT Megafunction Output Ports

Port Name	Required	Description	Comments
cal_shift_busy	Yes	Specifies the status of calibration or shifting operation. This signal is asserted until the calibration or shifting operation is completed.	Output port [OCT_BLOCK_NUMBER - 1..0] wide
calibration_busy	Yes	Specifies the status of the calibration operation. This signal is asserted until the calibration operation is completed.	Output port [OCT_BLOCK_NUMBER - 1..0] wide
paralleltermination control	Yes	Specifies parallel termination	Receives the current state of the pull-up and pull-down transmitter control buses from a termination logic block. Output port [OCT_BLOCK_NUMBER * 13..0] wide
seriesterminationcontrol	Yes	Specifies the serial termination	Receives the current state of the pull-up and pull-down receiver control buses from a termination logic block. Output port [OCT_BLOCK_NUMBER * 13..0] wide

This chapter provides additional information about the document and Altera.

Document Revision History

The following table lists the revision history for this document.

Date	Version	Changes
February 2012	3.0	Updated the following sections: ■ "Device Support" section ■ "MegaWizard Parameter Settings" section.
November 2008	2.0	■ Updated the following sections: ■ "Device Support" section ■ "Features" section ■ "Functional Description" chapter ■ "Design Example: Series Calibration for Four Calibration Blocks Using Stratix III Device" section ■ "Functional Simulation in the ModelSim-Altera Software" section ■ "This chapter describes the functional description and the design examples of the ALTOCT megafunction. This section also includes the ports descriptions of the ALTOCT megafunction. You can use the ports to customize the ALTOCT megafunction according to your application." section ■ "How to Contact Altera" section ■ Removed the following sections: ■ "Resource Utilization & Performance" section ■ "Software and System Requirements" section ■ "Instantiating Megafunctions in HDL Code" section ■ "Identifying a Megafunction after Compilation" section ■ "SignalTap II Embedded Logic Analyzer" section ■ Removed all screenshots in the "This section describes the parameter settings for the ALTOCT megafunction.You can parameterize the megafunction using the MegaWizard™ Plug-In Manager or the command-line interface (CLI). Altera recommends that you configure the megafunctions using the MegaWizard Plug-In Manager." section ■ Reorganized the "This section describes the parameter settings for the ALTOCT megafunction.You can parameterize the megafunction using the MegaWizard™ Plug-In Manager or the command-line interface (CLI). Altera recommends that you configure the megafunctions using the MegaWizard Plug-In Manager." section into table format. ■ Renamed "About this User Guide" section to "Additional Information" and moved the section to the end of the user guide.
December 2006	1.0	Initial release

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix n denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.